

·维数组的存储结构

ElemType a [10]: //ElemType型一维数组

C语言定义 ·维数组



起始地址: LOC

各数组元素大小相同, 且物理上连续存放。

数组元素a[i] 的存放地址 = LOC + i * sizeof(ElemType) (0≤i<10)

注:除非题目特别说明,否则数组下标默认从0开始

注意审题! 易错!

二维数组的存储结构

ElemType b[2][4]; //2行4列的二维数组

二维数组

b[0][0]	b[0][1]	b[0][2]	b[0][3]
b[1][0]	b[1][1]	b[1][2]	b[1][3]

逻辑视角

内 b[0][1] b[0][2] b[0][3] b[1][0] b[1][1] b[1][2] b[0][0] b[1][3] 存

行优先 存储

内 存

b[1][0] b[0][1] b[1][1] b[0][2] b[1][2] b[0][3] b[1][3] b[0][0]

列优先 存储

二维数组的存储结构

ElemType b[2][4]; //2行4列的二维数组

二维数组

b[0][0]	b[0][1]	b[0][2]	b[0][3]
b[1][0]	b[1][1]	b[1][2]	b[1][3]

逻辑视角

内 存

b[0][1] b[0][2] b[0][3] b[1][0] b[1][1] b[1][2] b[0][0]

b[1][3]

行优先 存储

起始地址: LOC

M行N列的二维数组 b[M][N] 中,若按<mark>行优先</mark>存储,则

b[i][j] 的存储地址 = LOC + (i*N + j) * sizeof(ElemType)

二维数组的存储结构

ElemType b[2][4]; //2行4列的二维数组

二维数组

b[0][0]	b[0][1]	b[0][2]	b[0][3]
b[1][0]	b[1][1]	b[1][2]	b[1][3]

逻辑视角

内 b[1][0] b[0][1] b[1][1] b[0][0] b[0][2] b[1][2] b[0][3] b[1][3]

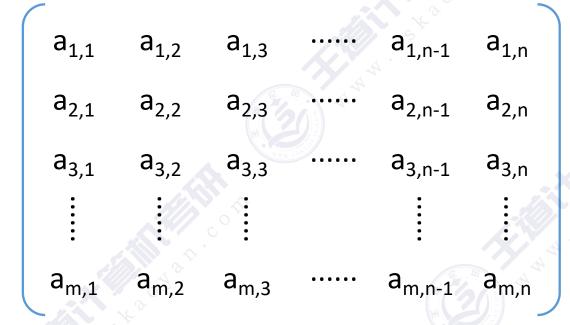
列优先 存储

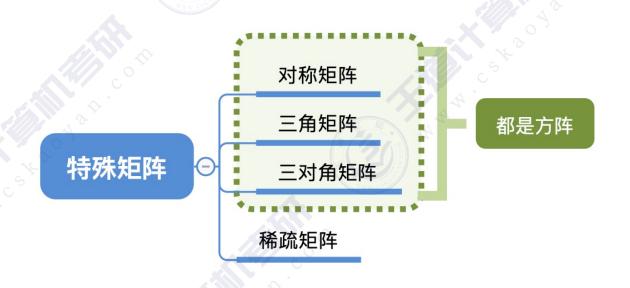
起始地址: LOC

M行N列的二维数组 b[M][N] 中,若按<mark>列优先</mark>存储,则

b[i][j] 的存储地址 = LOC + (j*M+ i) * sizeof(ElemType)

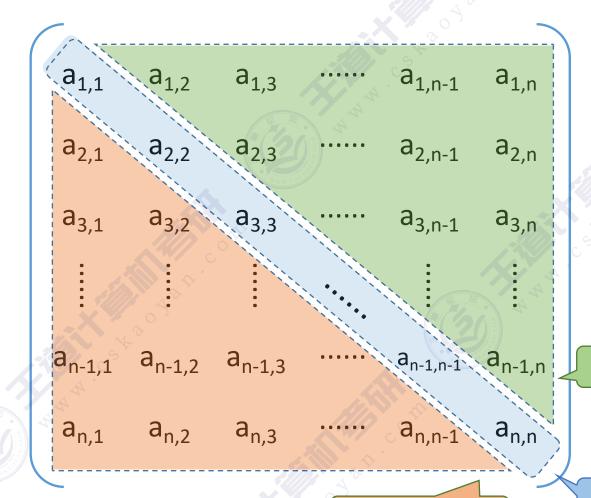
普通矩阵的存储





可用二维数组存储

注意:描述矩阵元素时,行、列号通常从1开始;而描述数组时通常下标从0开始 (具体看题目给的条件,注意审题!) 某些特殊矩阵可以压缩存储空间



若 n 阶<mark>方阵</mark>中任意一个元素 a_{i,j} 都有 a_{i,j} = a_{j,i} 则该矩阵为<mark>对称矩阵</mark>

普通存储: n*n 二维数组

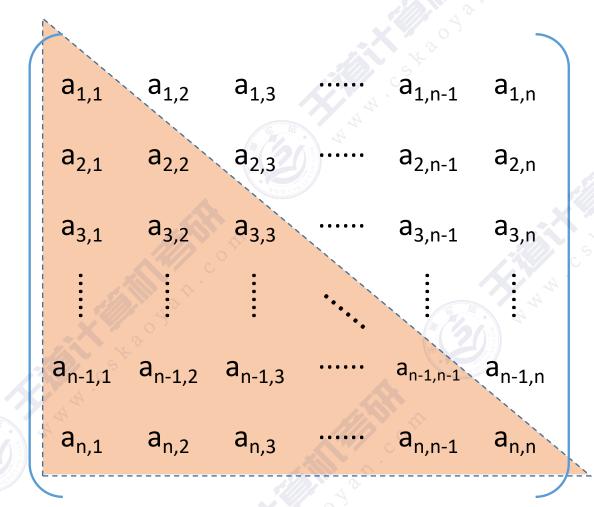
压缩存储策略: 只存储主对角线+下三角区

(或主对角线+上三角区)

上三角区: i<i

角区: i>j

主对角线: i=j



策略: 只存储主对角线+下三角区

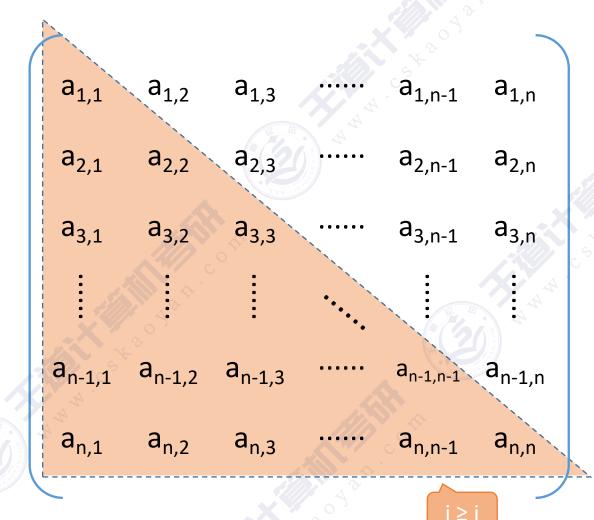
按行优先原则将各元素存入一维数组中。

B[0]	B[1]	B[2]	B[3]	····		B[?]
a _{1,1}	a _{2,1}	a _{2,2}	a _{3,1}	II WW CS	a _{n,n-1}	a _{n,n}



思考:

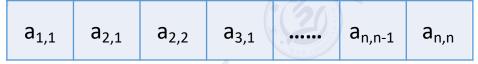
- ①数组大小应为多少?
- ②站在程序员的角度,对称矩阵压缩存储后怎样才能方便使用?
- ① (1+n)*n/2
- ②可以实现一个"映射"函数 矩阵下标→一维数组下标

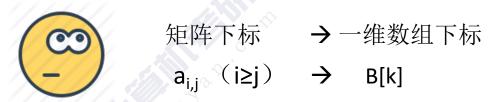


策略: 只存储主对角线+下三角区

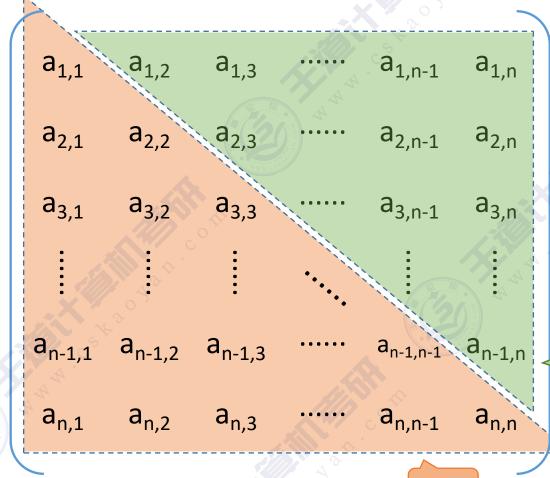
按行优先原则将各元素存入一维数组中。

B[0] B[1] B[2] B[3]
$$B[\frac{n(n+1)}{2}-1]$$





Key: 按<mark>行优先</mark>的原则,a_{i,j}是第几个元素?



策略: 只存储主对角线+下三角区

按行优先原则将各元素存入一维数组中。

B[0] B[1] B[2] B[3] $B[\frac{n(n+1)}{2}-1]$

$$a_{1,1} \quad a_{2,1} \quad a_{2,2} \quad a_{3,1} \quad \dots \quad a_{n,n-1} \quad a_{n,n}$$

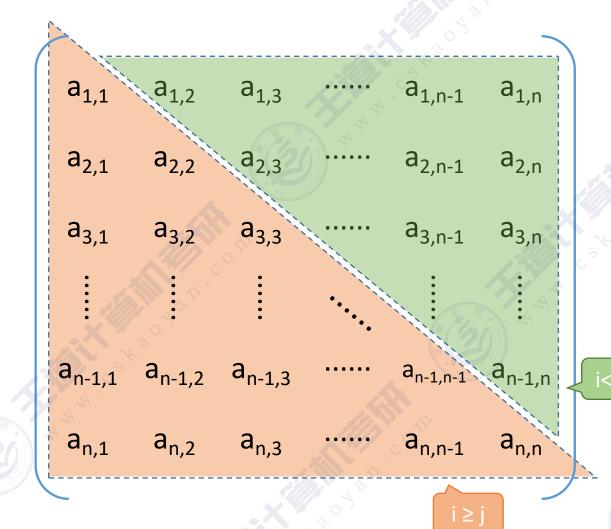


矩阵下标 → 一维数组下标

$$a_{i,j}$$
 (i\rightarrow B[k]

「<mark>a_{i,j}=a_{j,i}(对称矩阵性质)</mark>

$$\rightarrow$$
 k= $\frac{j(j-1)}{2} + i - 1$



策略: 只存储主对角线+下三角区

按行优先原则将各元素存入一维数组中。

B[0] B[1] B[2] B[3] B[
$$\frac{n(n+1)}{2}$$
-1]

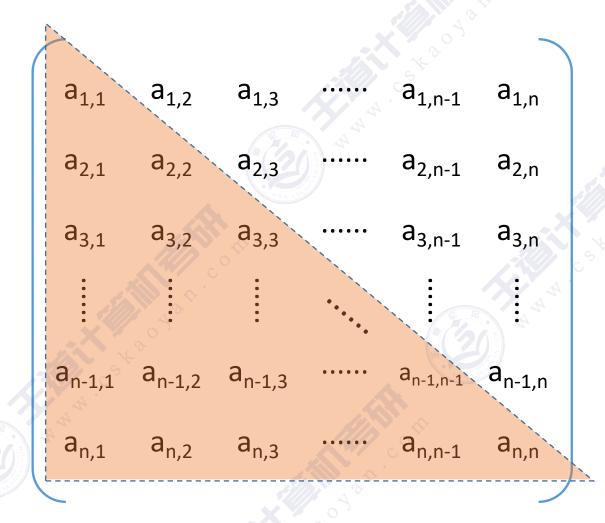
$$a_{1,1} \quad a_{2,1} \quad a_{2,2} \quad a_{3,1} \quad \dots \quad a_{n,n-1} \quad a_{n,n}$$

矩阵下标 → 一维数组下标

$$a_{i,j} \rightarrow B[k]$$

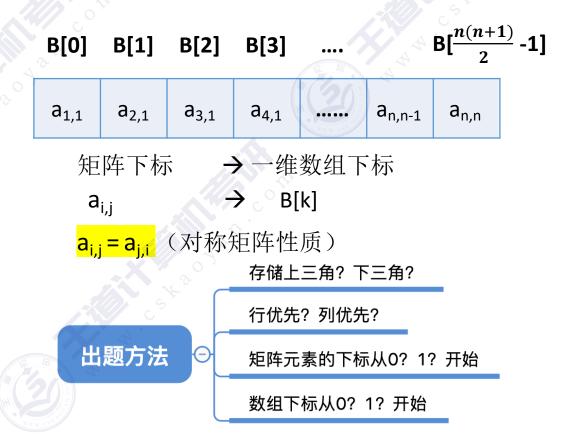
a_{i,j} = a_{i,i}(对称矩阵性质)

$$k = \begin{cases} \frac{i(i-1)}{2} + j - 1, & i \ge j \text{ (下三角区和主对角线元素)} \\ \frac{j(j-1)}{2} + i - 1, & i < j \text{ (上三角区元素} a_{ij} = a_{ji} \text{)} \end{cases}$$

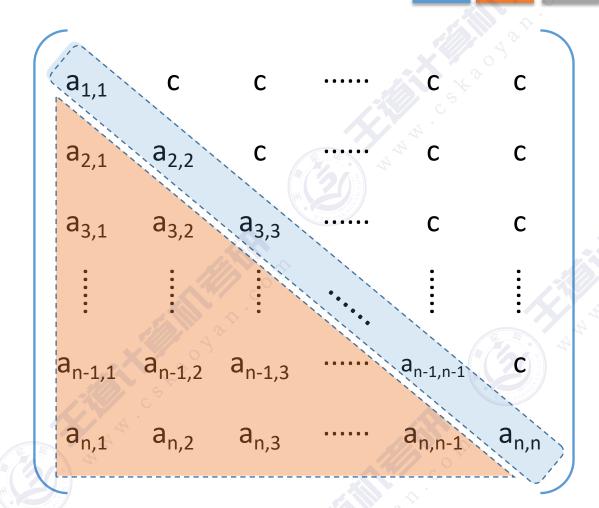


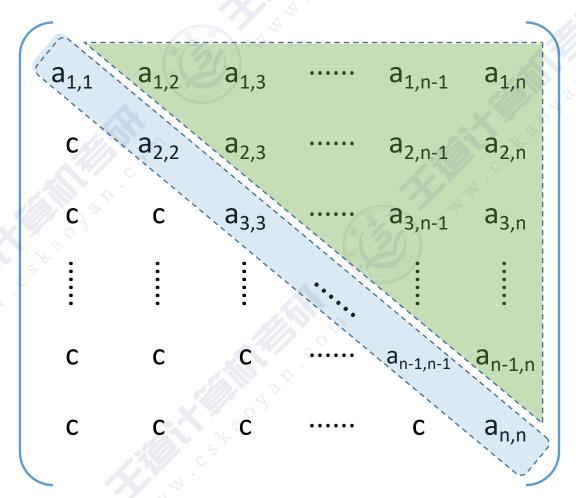
策略: 只存储主对角线+下三角区

按列优先原则将各元素存入一维数组中。



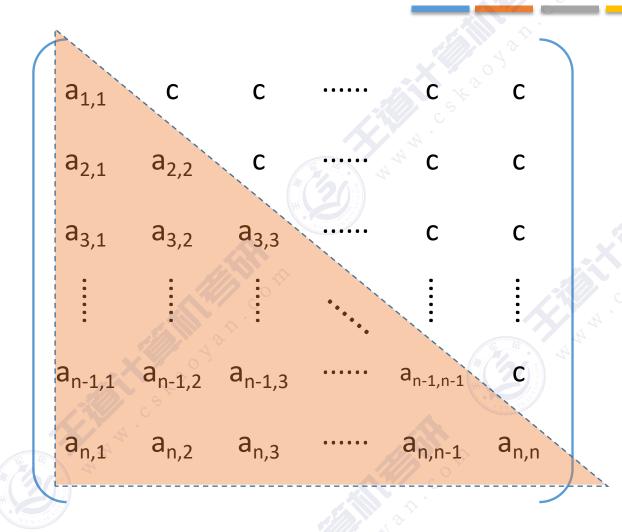
王道考研/CSKAOYAN.COM





下三角矩阵: 除了主对角线和下三角区,其余的 元素都相同

上三角矩阵:除了主对角线和上三角区,其余的 元素都相同

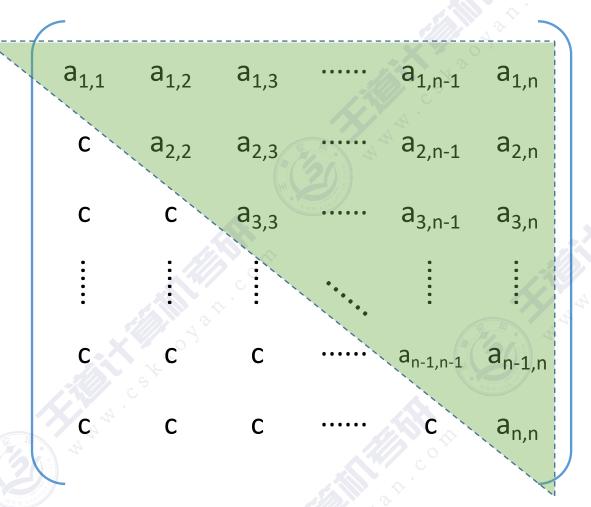


下三角矩阵:除了主对角线和下三角区,其余的 元素都相同 压缩存储策略:按<mark>行优先</mark>原则将橙色区元素 存入一维数组中。并<mark>在最后一个位置存储常</mark> 量c



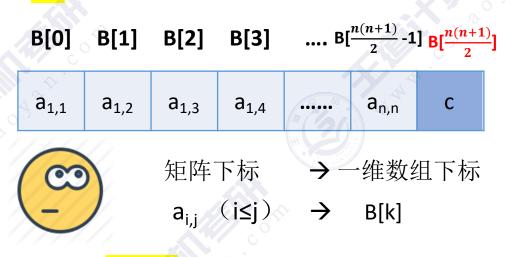
Key: 按<mark>行优先</mark>的原则,a_{i,j}是第几个元素?

$$k = \begin{cases} \frac{i(i-1)}{2} + j - 1, & i \ge j \text{ (下三角区和主对角线元素)} \\ \frac{n(n+1)}{2}, & i < j \text{ (上三角区元素)} \end{cases}$$



上三角矩阵:除了主对角线和上三角区,其余的 元素都相同

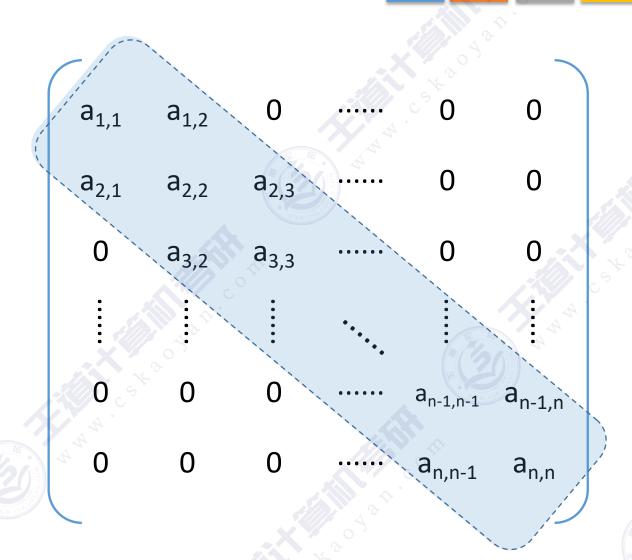
压缩存储策略:按<mark>行优先</mark>原则将绿色区元素 存入一维数组中。并<mark>在最后一个位置存储常量c</mark>



Key: 按<mark>行优先</mark>的原则,a_{i,j}是第几个元素?

$$k = \begin{cases} \frac{(i-1)(2n-i+2)}{2} + (j-i), & i \leq j \text{ (上三角区和主对角线元素)} \\ \frac{n(n+1)}{2}, & i > j \text{ (下三角区元素)} \end{cases}$$

王道考研/CSKAOYAN.COM



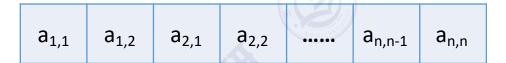
三对角矩阵,又称带状矩阵:

当 |i - j|>1 时,有 a_{i,j} = 0 (1≤ i, j ≤n)

压缩存储策略:

按<mark>行优先</mark>(或列优先)原则,只存储带状部分

B[0] B[1] B[2] B[3] B[3n-3]





矩阵下标

→一维数组下标

 $a_{i,j}$ (|i-j| \leq 1) \rightarrow

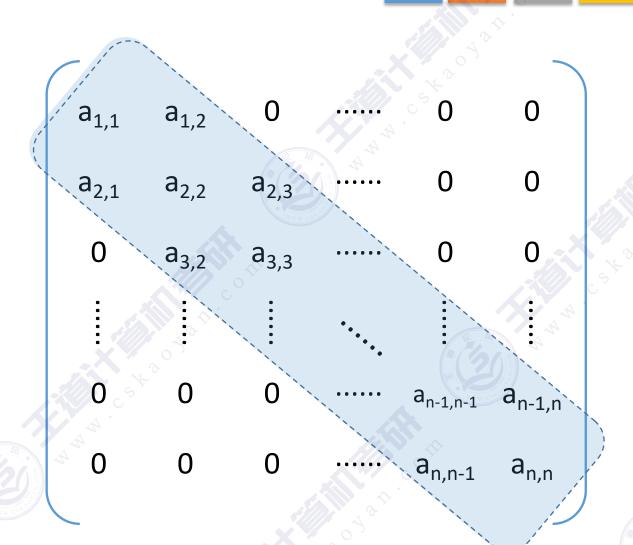
B[k]

Key: 按<mark>行优先</mark>的原则,a_{i,i}是第几个元素?

前i-1行共 3(i-1)-1 个元素 a_{i,j}是 i 行第 j-i+2 个元素 a_{i,i}是第 2i+j-2 个元素



 \rightarrow k = 2i+j-3





若已知数组下标k,如何得到i,j?

B[k]

 \rightarrow $a_{i,j}$

第 k+1 个元素,在第几行?第几列?

前i-1行共 3(i-1)-1 个元素 前i行共 3i-1 个元素 显然, 3(i-1)-1 < k+1 ≤ 3i-1

i≥(k+2)/3 一 可以理解为"刚好"大于等于

i = [(k+2)/3] 向上取整即可满足 "刚好"大于等于

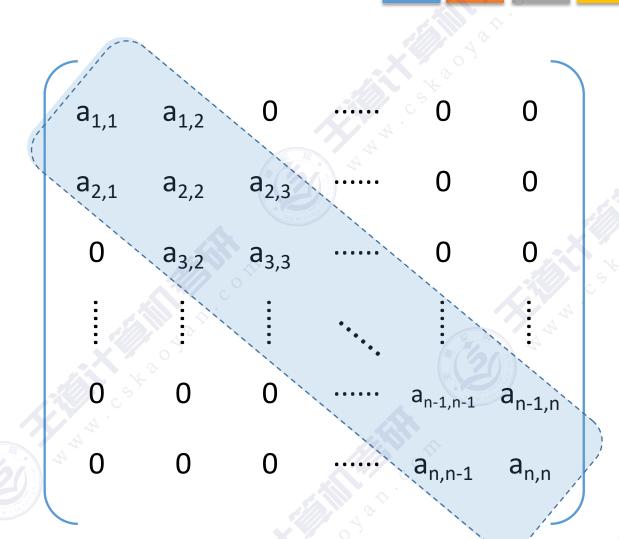
王道书的计算逻辑: 3(i-1)-1 ≤ k < 3i-1

 $i \le (k+1)/3+1$

可以理解为"刚好"小于等于

i = [(k+1)/3+1]

向下取整即可满足 "刚好"小于等于

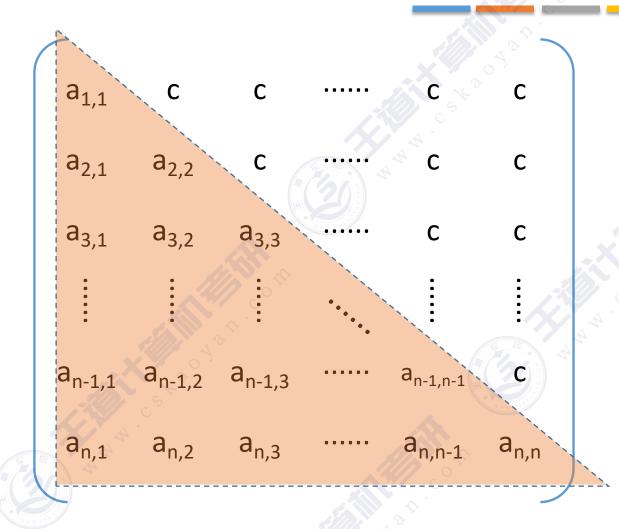




若已知数组下标k,如何得到i,j?

 $B[k] \rightarrow a_{i,j}$

第 k+1 个元素,在第几行?第几列?



下三角矩阵:除了主对角线和下三角区,其余的 元素都相同 压缩存储策略:按<mark>行优先</mark>原则将绿色区元素 存入一维数组中。并<mark>在最后一个位置存储常</mark>量c



Key: 按<mark>行优先</mark>的原则,a_{i,j}是第几个元<u>出i</u>, j?

$$k = \begin{cases} \frac{i(i-1)}{2} + j - 1, & i \ge j \text{ (下三角区和主对角线元素)} \\ \frac{n(n+1)}{2}, & i < j \text{ (上三角区元素)} \end{cases}$$

王道考研/CSKAOYAN.COM

稀疏矩阵的压缩存储

0	0	4	0	0	5
0	3	0	9	0	0
0	0	0	0	7	0
0	2	0	0	0	0
0	0	0	0	0	0

稀疏矩阵: 非零元素远远少于矩阵元素的个数

压缩存储策略:

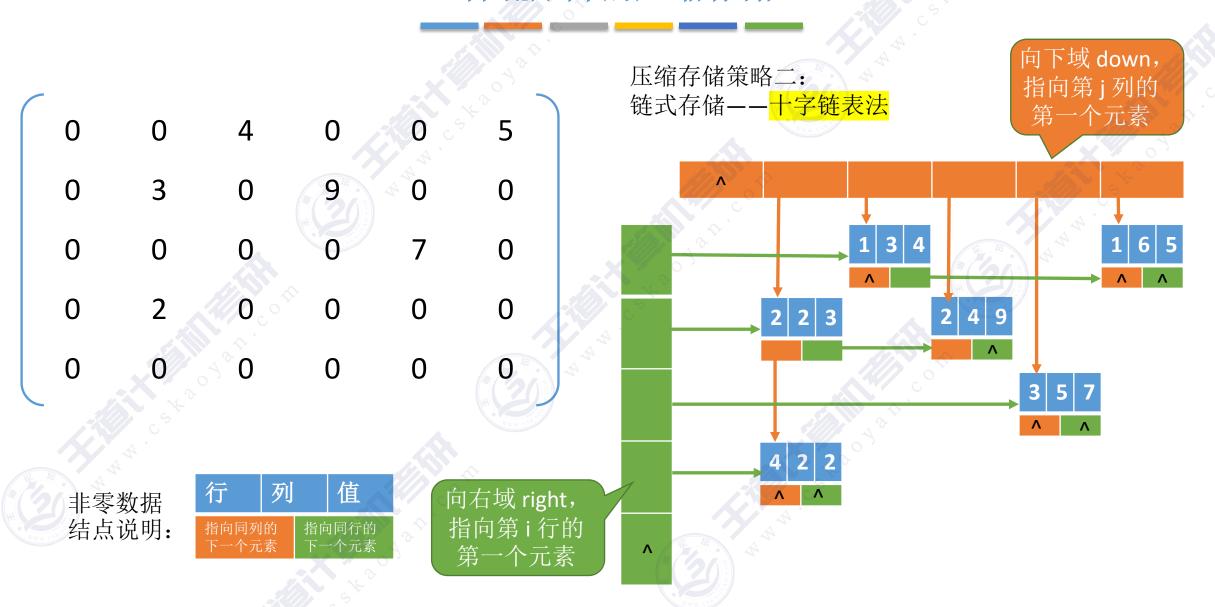
顺序存储——三元组 <行,列,值>

i (行)	j (列)	v(值)
1	3	4
1	6	5
2	2	3
2	4	9
3	5	7
4	2	2

(注:此处行、列标从1开始)



稀疏矩阵的压缩存储



知识回顾与重要考点

特点:对方阵中的任意一个元素,有 ai,j = aj,i

压缩:只存储主对角线+下三角区(或主对角线+上三角区)

特点:上三角区全为常量(下三角矩阵);或下三角区全为常量(上三角矩阵)

压缩:按行优先/列优先规则依次存储非常量区域,并在最后一个位置存放常量c

特殊矩阵 压缩存储 三对角矩阵 (带状矩阵)

三对角矩阵 特点: 当 | i - j| > 1 时,有 ai,j = 0 (1≤ i, j ≤ n)

压缩:只存储非零元素

压缩:按行优先/列优先规则依次存储带状区域

非零元素个数远小于零元素个数

稀疏矩阵

对称矩阵

三角矩阵

顺序存储: 顺序存储三元组 <行,列,值>

链式存储: 十字链表法

知识回顾与重要考点

矩阵的压缩存储需要多长的数组

由矩阵行列号 <i, j> 推出对应的数组下标号 k

数列 求和

如何处理不等式中的"刚好大于等于/小于等于"

由 k 推出 <i,j>

易错点

向上取整/向下取整

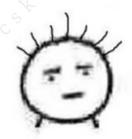
存储上三角? 下三角?

行优先? 列优先?

矩阵元素的下标从0? 1? 开始

数组下标从O? 1? 开始

常见考题



问题

是否忘了, 等差数列求和???

欢迎大家对本节视频进行评价~



学员评分: 3.4 特殊矩...







🚫 公众号: 王道在线



b站:王道计算机教育



→ 抖音: 王道计算机考研

